

INFORMATICS PRACTICES

CLASS XII

[As per CBSE Curriculum 2024-25]

The Python logo, consisting of two interlocking snakes, one blue and one yellow, is positioned in the upper center of the page.

Python
for beginners

**QUICK
REVISION**

CLASS - XII: INFORMATICS PRACTICES (2024-25)

QUICK REVISION

UNIT-I [25 Marks]	1 M	2 M	3 M	4 M	5M	Total
Data Handling using Pandas and Data Visualization	9	4	3	4	5	25 M

Introduction to Python Libraries-

A Python library or package is a collection of Python modules containing ready-to-use functions to perform specific task related to an application. The advantage of using libraries is that we can import libraries and can use its different functions in Python program. Some commonly used Python Libraries are-

1. **Python Standard Library:** It is a collection of modules which is available after installing Python. Examples of such libraries are – (a) math module- provides mathematical functions (b) random module- provides functions for generating random numbers. (c) statistics module- provides statistical functions
2. **Numpy (Numerical Python) library:** Offers functions for working with matrices and multi-dimensional arrays.
3. **Pandas (PANel + DAta) library:** Pandas is a fast, powerful, flexible and easy to use open-source library for data manipulation and analysis of data. Commonly used to handle Series and Data Frame.
4. **Matplotlib library:** It provides functions for Visualization of Data i.e. plotting and drawing graphs.

A Python library can be imported in a program by using import command as –
import <library Name> [as <object>]

Ex: import pandas as pd
import numpy as np

What is PANDAS?

Pandas (**PANel DAta System**) is fast and powerful open source data analysis library of Python developed by Wes McKinney. Pandas has functions for analysing, cleaning, exploring, and manipulating massive data.

Features of Pandas:

- Pandas Data structures are **Fast and efficient in functionality.**
- Supports functions to handle **missing data.**
- Supports **Label-based slicing and indexing.**
- Offers **conditional selection, merging and joining of data sets.**
- Offers import/export to handle variety of data source.

Data Structure in PANDAS:

A data structure is a collection of data values and operations that can be applied to that data. It enables efficient storage, retrieval and modification of data.

Pandas offer the Series and Data Frame data structures for handling and analysis of big data.

Series:

- Series is one dimensional (1D) homogeneous data structure with labelled index.
- Series is size immutable and Value Mutable i.e. new values cannot be added but existing values can be deleted and modified.
- The default positional index starts from zero, if index/label are not given.

DataFrame:

- Data Frame is two-dimensional (2D) heterogenous data structure with rows and columns like Excel sheet.
- Data Frame is both Size and Value mutable i.e. you can add new rows /columns or delete.
- Rows and Column of Data Frame are indexed or labelled with row index and column index. Default positional index starts with 0, if indexes are not given.

<table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="width: 10%;"></th> <th style="width: 10%;">Data</th> </tr> </thead> <tbody> <tr><td style="text-align: center;">0</td><td style="text-align: center;">2.0</td></tr> <tr><td style="text-align: center;">1</td><td style="text-align: center;">NaN</td></tr> <tr><td style="text-align: center;">2</td><td style="text-align: center;">2.0</td></tr> <tr><td style="text-align: center;">3</td><td style="text-align: center;">-3.0</td></tr> <tr><td style="text-align: center;">4</td><td style="text-align: center;">-3.0</td></tr> </tbody> </table> <p style="text-align: center;">Series</p>		Data	0	2.0	1	NaN	2	2.0	3	-3.0	4	-3.0	<table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="width: 10%;"></th> <th style="width: 30%;">Animal</th> <th style="width: 60%;">Number_legs</th> </tr> </thead> <tbody> <tr><td style="text-align: center;">0</td><td style="text-align: center;">lion</td><td style="text-align: center;">4.0</td></tr> <tr><td style="text-align: center;">1</td><td style="text-align: center;">fox</td><td style="text-align: center;">4.0</td></tr> <tr><td style="text-align: center;">2</td><td style="text-align: center;">cow</td><td style="text-align: center;">4.0</td></tr> <tr><td style="text-align: center;">3</td><td style="text-align: center;">spider</td><td style="text-align: center;">8.0</td></tr> <tr><td style="text-align: center;">4</td><td style="text-align: center;">snake</td><td style="text-align: center;">NaN</td></tr> </tbody> </table> <p style="text-align: center;">DataFrame</p>		Animal	Number_legs	0	lion	4.0	1	fox	4.0	2	cow	4.0	3	spider	8.0	4	snake	NaN
	Data																														
0	2.0																														
1	NaN																														
2	2.0																														
3	-3.0																														
4	-3.0																														
	Animal	Number_legs																													
0	lion	4.0																													
1	fox	4.0																													
2	cow	4.0																													
3	spider	8.0																													
4	snake	NaN																													

Creating Series:

A series can be created using **Series()** function with optional parameters for data and index.
<Series Object>= pandas.Series([data=<data set>], [index=<index set>], [dtype=<datatype>])
Where *dataset* can be a list/Dictionary or any scalar value.

Creating Empty Series:

An empty series can be created using Series () without any parameters.

```
import pandas as pd
s= pd.Series()
print(s)
```

Output:
Series([], dtype: float64)

Creating Series with ndarray:

A series can be created using numPy array (ndArray), as data in Series() method. Default index [0,1,2..] will be generated, if labeled index is not given.

```
import pandas as pd
import numpy as np
arr=np.array([10,20,30,40])
s= pd.Series(arr)
print(s)
```

0 10
1 20
2 30
3 40

<p>Creating Series with list: A series can be created by passing a list as data in Series() method. If no index is given then by default index will be generated as [0,1,2,3...len(list)-1].</p>	<pre>import pandas as pd lst=[5,6,7,8] idx=['p','q','r','s'] s=pd.Series(lst,index=idx) print(s)</pre>	<p>p 5 q 6 r 7 s 8</p>
<p>Creating Series with Dictionary: A series can be created using dictionary as data in Series() method. The keys of dictionary become index and values are assumed as data. Note: If index is provided then Series created with matching index and Keys only. NaN (Not a number) value is used for not matching index.</p>	<pre>import pandas as pd d={'A':45,'B':67,'C':34,'D':78} s= pd.Series(d) print(s)</pre>	<p>A 45 B 67 C 34 D 78</p>
	<pre>import pandas as pd d={'A':45,'B':67,'C':34,'D':78} idx=['A','S','D','C'] s= pd.Series(d,index=idx) print(s)</pre>	<p>A 45.0 S NaN D 78.0 C 34.0</p>
<p>Creating Series with Scalar value: A series can be created using a constant (Scalar) value. The value is repeated as per given index.</p>	<pre>import pandas as pd s1=pd.Series(8) print(s1) s2=pd.Series(5,index=['a','b','c']) print(s2)</pre>	<p>0 8 ----- a 5 b 5 c 5</p>
<p>Accessing Elements of Series: Data values of a series can be accessed in three ways-</p> <ul style="list-style-type: none"> • Positional Index: Single value at given index. • Labelled Index: Single value at given index. • Slicing: To assess multiple values (subset) for given range of indexes. <p>The range can be defined as [start : end : step]. Default step value is 1. Negative step value will cause access of series in reverse order.</p> <p>Note: Labeled index : values are retrieved from START to END Positional index: values are retrieved from START to END-1</p>	<pre>import pandas as pd val=[10,20,30,40,50,60] idx=['a','b','c','d','e','f'] s=pd.Series(val,index=idx) # using positional index print (s[1]) # using labelled index print(s['b']) # slicing using Positional index print (s[1:5]) # Slicing using labelled index print(s['b':'d'])</pre>	<p>20 ----- 20 ----- b 20 c 30 d 40 e 50 ----- b 20 c 30 d 40</p>
<p>Conditional Access of Series elements: You can filter/access data values of a series based on defined condition. Applying condition on whole Series: It will return True or False based on given condition. Applying condition on elements: It will returns selected values based on condition. Any single conditional expression with Relational operators (>,<=,!=,>=,<=) can be applied on series.</p>	<pre>import pandas as pd val=[5,20,10,80,25] idx=['a','b','c','d','e'] s=pd.Series(val,index=idx) # applying condition on whole series print(s>20) # applying condition on elements print (s[s>20])</pre>	<p>a False b False c False d True e True ----- d 80 e 25</p>
<p>Modifying Series Elements : You can modify data value of corresponding index by providing index/ position. To modify values in a range of indexes, slicing can be used.</p> <p>Series[index] = <new value> Series[start : end] = <new value></p> <p>Modifying Index of Series: Index of Series can be modified using index attribute of series. Series.index = <new index values></p>	<pre>import pandas as pd lst=[10,20,30,40,50] idx=['a','b','c','d','e'] s=pd.Series(lst,index =idx) # modifyng single value s['a']=100 # modifying multiple values s['c':'e']=5 print(s) #modifying index s.index=['p','q','r','s','t'] print(s)</pre>	<p>a 100 b 20 c 5 d 5 e 5 ----- p 100 q 20 r 5 s 5 t 5</p>

Series Attributes:		<pre>import pandas as pd import numpy as np lst=[4, 5, np.NaN, 7, 8, 9] idx=['a', 'b', 'c', 'd', 'e', 'f'] s= pd.Series(lst, index=idx) s.name="MySeries" print(s) print(s.size) print(s.shape) print(s.hasnans) print(s.dtype) print(s.empty) print(s.values) print(s.index)</pre>	<pre>a 4.0 b 5.0 c NaN d 7.0 e 8.0 f 9.0 Name: MySeries, dtype: float64 6 (6,) True float64 False [4. 5. nan 7. 8. 9.] Index(['a','b','c','d','e','f'], dtype='object')</pre>
Attribute	Purpose		
name	Used to assign name to Series.		
Size	Returns size (number of elements) of series		
Shape	Returns shape of series as tuple		
values	Returns data values as ndarray		
index	Returns index labels		
Dtype	Returns data type of series		
Empty	Returns true is series is empty otherwise false		
Hasnans	Returns true if series has NaN values.		

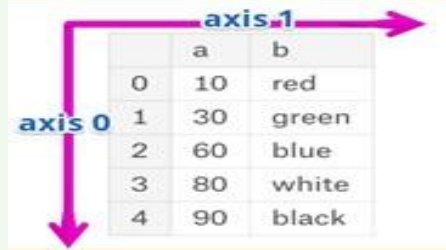
Series Methods:		<pre>import pandas as pd lst=[4,5,6,7,8,9] idx=['a','b','c','d','e','f'] s= pd.Series(lst, index=idx) print(s.count()) print(s.head(3)) print(s.tail(2)) print(s.min()) print(s.max()) print(s.sum()) print(s.add(5))</pre>	<pre>6 a 4 b 5 c 6 ----- e 8 f 9 ----- 4 9 39 a 9 b 10 c 11 d 12 e 13 f 14</pre>
Series methods specific operations on Series (s).			
head()	Returns top 5 values of series, if no value is given.		
tail()	Returns bottom 5 values of series, if no value is given.		
count()	Returns counting of not-NaN values in series.		
<i>The following Mathematical methods are applicable on numeric series only.</i>			
min()	Returns minimum value of the series.		
max()	Returns maximum value of the series.		
sum()	Returns total of value of the series.		
add()	Adds a scalar (constant) value or another series.		
sub()	Subtracts a scalar (constant) value or another series.		
mul()	Multiplies series with scalar (constant) values or another series.		
div()	Divides series with scalar (constant) values or another series.		

Mathematical operations on Series:		<pre>import pandas as pd lst=[4,5,6,7,8] idx=['a','b','c','d','e'] s= pd.Series(lst, index=idx) #applying arithmetic operation print(s.add(5)) # OR print(s+5)</pre>	<pre>a 9 b 10 c 11 d 12 e 13</pre>	
<p>Mathematical operations on series can be applied in two ways.</p> <p>Using operators : +, -, *, /, //, % etc.</p> <p>Using Methods: add(), sub(), mul(), div() etc.</p> <p>Mathematical operations can be two types-</p> <p>Manipulating Series with scalar (constant) value:</p> <p>When a series is manipulated with a constant number then mathematical operation is applied with each element of the series (Vector arithmetic).</p> <p>Manipulating two Series:</p> <p>When mathematical operation is applied on two series then operation is performed on matching index. The NaN value will be produced for non-matching/missing value.</p> <p>Note: We can use <i>fill_value</i> parameter to avoid NaN result for non-matching values.</p>				
<pre>import pandas as pd s1= pd.Series([4,5,6,7], index=['a','b','c','d']) s2= pd.Series([10,12,15,18,20], index=['a','p','b','q','c']) #applying arithmetic operation print(s1.add(s2)) #OR (s1+s2)</pre>				<pre>a 14.0 b 20.0 c 26.0 d NaN p NaN q NaN</pre>
<p>#applying arithmetic operation</p> <p># 0 will be assumed for non-matching values # before adding series.</p> <pre>s1.add(s2, fill_value=0)</pre>				<pre>a 14.0 b 20.0 c 26.0 d 7.0 p 12.0 q 18.0</pre>

Data Frame:

A Data Frame is 2-dimensional heterogeneous data structure arranged in tabular form containing rows and columns. Each row and column may have labeled (Text) index as well as Positional (Numeric) Index.

The default numeric index labels starts from zero, if no index is given. Dimensions of DataFrame are also called **Axis**. DataFrame is **Size and Data mutable**.



Creating Data Frame:

Data Frame can be created in the following two main approaches.

(Row wise) List with nested structure.	List-List	[[R1],[R2],[R3],...]
	List-Dictionary	[{..},{..},{..},...]
	List-Series	[S1,S2,S3,...]
(Column wise) Dictionary with nested structure.	Dictionary-List	{ [C1],[C2],[C3],... }
	Dictionary -Dictionary	{ {..},{..},{..},... }
	Dictionary -Series	{ S1,S2,S3,... }

A Data Frame in Pandas can be created using DataFrame() function with optional parameters for **data** , **row index** and **column index**.

<DF Object>= pandas.DataFrame(<2D-data set>, index=<Row indexes>, columns=<column indexes>)

Where 2D-dataset may be ndarray or nested structure of list of dictionary.

Data Frame can also be created using ndarray or another Data Frame.

Creating Empty DataFrame:

```
import pandas as pd
df= pd.DataFrame()
print(df)
```

Creating Data Frame using List of Lists (Nested List):

```
import pandas as pd
lst=[['Amar',60,68,45], ['Akbar',65,65,56], ['Anthony',70,77,65]]
df= pd.DataFrame(lst,index=[1,2,3], columns=['Name','Phy','Chem','Maths'])
print(df)
```

Creating Data Frame with list of Dictionary:

```
import pandas as pd
lst= [ {'Name':'Amar','Phy':60,'Chem':68,'Maths':45}, {'Name':'Akbar','Phy':65,'Chem':65,'Maths':56},
      {'Name':'Anthony','Phy':70,'Chem':77,'Maths':65} ]
df= pd.DataFrame(lst, index=[1,2,3])
print(df)
```

Creating Data Frame with Dictionary of Series

```
s1=pd.Series(['Amar','Akbar','Anthony'],index=[1,2,3])
s2=pd.Series([60,65,70], index=[1,2,3] )
s3=pd.Series([68,65,77], index=[1,2,3] )
s4=pd.Series([45,56,65], index=[1,2,3] )
dct={'Name':s1,'Phy':s2,'Chem':s3,'Maths':s4}
df= pd.DataFrame(dct)
print(df)
```



	Name	Phy	Chem	Maths
1	Amar	60	68	45
2	Akbar	65	65	56
3	Anthony	70	77	65

Note: Keys of dictionary become column index and values arranged as per matching keys)

Accessing Data Frame:

The values of data frame can be accessed in different ways like Series. The possible ways are-

- Accessing column(s)
- Accessing Row(s)
- Accessing Row(s) and column(s)

	Name	Phy	Chem	Maths
one	Amar	60	68	45
two	Akbar	65	65	56
three	Anthony	70	77	65

Accessing Columns of DataFrame:

Single column:

<DataFrame>[<Col Label>] OR <DataFrame>.<Col Label>

Multiple Columns: **<DataFrame>[[Col 1, Col2, Col3..]]**

```
print(df[ 'Phy' ]) OR
print(df.Phy)
print(df[[ 'Phy', 'Maths' ]])
Print(df[[ 'Maths', 'Phy' ]])
```

Accessing Rows of Data Frame:

Rows can be accessed by giving Index range.

<DataFrame>[<Start row label>:<End row label>]

When start or end label is not provided, default value first row and last row will be used.

When labeled index used the **START** to **END** will be displayed. In case of Positional index (number) then **START** to **END-1** will be displayed.

Pandas head() and tail() function can also be used.

from row one onward

```
print(df[ 'one' : ])
```

```
print(df[ : 'three' ]) # till three row
```

from row one to three

```
print(df[ 'one' : 'three' ])
```

Single Row 'Two' only

```
Print(df[ 'two' : 'two' ])
```

```
Print(df[1:3]) # using positional index
```

```
Print(df.head(2))
```

<p>Accessing Rows and Columns using Label index: Pandas <code>loc[]</code> provides subset of selected rows, columns or both using row/column labels. <u>It returns selected subset from given Row/Col label from Start to End.</u> <code><DF object>.loc[<start row>:<end row>, <start column>:<end column>]</code> When start/end row/column is not provided, default value first row/column and last row/column will be used.</p>	<p>Accessing Rows and columns using Positional Indexes. <code>iloc[]</code> method to get subset of selected combinations of rows, columns or both using positional indexes. <code><DF object>.iloc[<start row>:<end row>, <start column>:<end column>]</code> Note: <code>iloc()</code> used positional indexes and row/col will be selected from START to END-1.</p>
<pre># all rows and all columns print(df.loc[:, :]) # from row 'one' to end row and all columns print(df.loc['one':, :]) # till row 'three' and all columns print(df.loc[:'three', :]) # all rows and from 'Phy' column to end print(df.loc[:, 'Phy':]) # all rows and from 'phy' to 'Maths' column print(df.loc[:, 'Phy':'Maths']) # from Row 'Two' to 'three' and 'Phy' to 'Chem' col print(df.loc['two':'three', 'Phy':'Chem'])</pre>	<pre># all rows and all columns print(df.iloc[:, :]) # from row 'one' to end row and all columns print(df.iloc[0:, :]) # till row 'two' and all columns print(df.iloc[:2, :]) # all rows and from 'Phy' column to end print(df.iloc[:, 1:]) # all rows and from 'phy' to 'Chem' column print(df.iloc[:, 1:3]) # from Row 'two' to 'two' and 'Phy' to 'Chem' print(df.iloc[1:2, 1:3])</pre>
<p>Accessing Individual Value: There are three methods to access individual value. <code><DFobject>.<Column>[<row label or row position>]</code> <code><DFobject>.at[<row label>, <column label>]</code> <code><DFobject>.iat[<row position>, <column position>]</code></p>	<p># accessing value of individual cell <code>print(df.Maths['three'])</code> <code>print(df.Maths['three'])</code> <code>print(df.Maths[2])</code> <code>print(df.at['two', 'Chem'])</code> <code>print(df.iat[1,1])</code></p>
<p>Accessing Items using condition You can also filter and analyze data by displaying True/False based on given condition. The condition may have relational operators like <code>>, >=, <, <=, !=, =</code> etc.</p>	<p># applying condition on 'Phy' Column <code>print(df['Phy']>60)</code> # filtering data on condition <code>print(df[df['Phy']>60])</code> # applying condition on range <code>print(df.loc['one':'two', 'Phy':]>60)</code> <code>print(df.loc[df['Chem']>60, ['CS']])</code> <code>print(df.loc[df['Chem']>60, ['CS', 'Maths']])</code></p>
<p>Accessing Items based on Boolean Indexing: Boolean indexing as name suggests, having Boolean values (True/False or 1/0) as row index/label to access the rows based on True or False criteria. For this we have to create/change row labels as True or False.</p>	<pre>dct={'Name':['Amar', 'Akbar', 'Anthony', 'Manpreet'], 'Phy':[60,65,70,67], 'Chem':[34,55,32,46], 'Maths':[45,56,65,75]} df=pd.DataFrame(dct, index=[True, False, True, False]) print(df) # accessing rows based on Boolean value of rows. print(df.loc[True]) print(df.loc[False])</pre>
<p>Modifying Data Frame:</p>	
<p>Modifying values of Existing Column: Single Column: <code><DF object>[<Column Label>]=<new value(s)></code> Multiple Column <code><DF Object>.loc[:, Column 1: column 2]=<new value(s)></code> If given column already exists in the Data Frame then existing values will be changed otherwise a new column will be added in the Data Frame.</p>	<pre>#modifying value of existing column 'Maths' df['Maths']=50 #modifying value of 'Phy' to 'Chem' column df.loc[:, 'Phy': 'Chem'] = 45 #Adding new column 'Eng' with 65 marks df['Eng']=65 # Adding new column 'IP' with new marks df['IP']=[45, 54, 67]</pre>
<p>Adding New Column: <code><DF object>[<Column Label>]=<new value(s)></code> We can also use expressions while creating new columns.</p>	<pre>df['IP']=[45, 54, 67] df['Total']= df['Phy']+df['Chem']+df['Maths']</pre>
<p>Adding new Row: <code><DF Object>.loc[<Row index>]=<new value(s)></code></p>	<pre># Adding new row 'five' with different marks df.loc['five']=['Manpreet',45,54,67]</pre>

Modifying values of Rows:
`<DF Object>.loc[<Row 1> : <Row 2>]=<new value(s)>`
 Rows of Data Frame can be modified by selecting and assigning new values. If given row already exists then existing values will be changed otherwise a new row will be added.

```
df.loc['one']=50
#modifying value of 'two' to 'three' row
df.loc['two':'three',:]=60
#Adding new row 'four' with 65 marks
df.loc['four',:]=65
```

Modifying single value at specified position:
`<DFobject>.<Column>[<row label or position>]=<new value>`
`<DFobject>.at[<row label>, <column label>]=<new value>`

```
df.Chem['two']=10
df.Chem[0]=20
df.at['two','Phy']=30
df.iat[2,1]=40
```

Deleting Rows and Columns of Data Frame:
 Rows or column of DataFrame can be deleted by two methods-

del <DF object>[<column label>] # to delete a single column
<DF Object>.drop([<Row labels/Column labels>], axis=0/1)
Where axis=1 for column deletion and axis=0 for row deletion.
 Using drop() method, you have to assign modified Data Frame on same or different object, since drop() deletes on copied object.

```
# Deleting a column 'Maths'
del df['Maths']
# Deleting a row 'two'
df=df.drop(['two'])
#deleting multiple column
df=df.drop(['Phy','Chem'],axis=1)
# deleting multiple rows
df=df.drop(['one','four'], axis=0)
```

Renaming row and column labels:
 Row and column indexes can be rename as-
`<DF Object>.rename(index={name-dict}, columns={column-dict})`

Renaming Row labels
`df=df.rename(index={'P':'one','Q':'two','R':'three','S':'four'})`

Renaming column labels
`df=df.rename(columns={'A':'Name','B':'Phy','C':'Chem','D':'Maths'})`

Data Frame Attributes:
 Once Data Frame has been created, you can access certain properties of Data Frame through predefined attributes.
 You can access series attributes as `<DF>.<attribute>`

Attributes	Purpose	
index	Returns list of row indexes /labels of Data Frame.	
columns	Returns list of column indexes/labels of Data Frame.	
values	Returns data values of Data Frame as ndarray.	
size	Returns the size (number of data items) of DataFrame.	
shape	Returns shape (dimensionality) of Data Frame as tuple	
axes	Returns list of both axes (row and columns labels)	
dtype	Returns column wise data type of Data Frame.	
empty	Returns true if Data Frame is empty otherwise false.	
T	Transpose index and columns.	

```
import pandas as pd
dct={'Phy':[60,65,70], 'Chem':[34,55,32], 'Maths':[45,56,65]}
df=pd.DataFrame(dct, index=['Amar','Akbar','Anthony'])
print(df)
print(df.index)
print(df.columns)
print(df.values)
print(df.size)
print(df.shape)
print(df.dtypes)
print(df.empty)
print(df.T)
```

```

          Phy  Chem  Maths
Amar      60   34    45
Akbar     65   55    56
Anthony   70   32    65
Index(['Amar', 'Akbar', 'Anthony'], dtype='object')
Index(['Phy', 'Chem', 'Maths'], dtype='object')
[[60 34 45]
 | [65 55 56]
 | [70 32 65]]
9
(3, 3)
Phy      int64
Chem     int64
Maths    int64
dtype: object
False
          Amar  Akbar  Anthony
Phy      60     65     70
Chem     34     55     32
Maths    45     56     65
```

Data Frame Methods:
 Apart from mathematical methods add(), sub(), mul(), div() etc., Pandas offers other methods too, which can be used for different operations on the Data Frame.

Methods	Purpose	
head()	Returns top 5 rows of Data Frame, if no value is given.	<pre>print(df.count()) #OR print(df.count(axis=0)) print(df.count(axis=1)) print(df['Phy'].count()) print(df.sum()) #OR print(df.sum(axis=0)) print(df.sum(axis=1)) print(df['Phy'].sum()) print(df.max()) #OR print(df.max(axis=0)) print(df.max(axis=1)) print(df['Phy'].max())</pre>
tail()	Returns bottom 5 rows, if no value is given.	
len(Df)	Return numbers of rows in the Data Frame.	
<i>The following functions can be applied in Row-wise/Column wise by specifying axis or whole data frame.(axis=1 for Row-wise and axis=0 for column wise). If axis is not given then 0 will be assumed.</i>		
count()	Returns counting of not-NaN values in Data Frame.	
min()	Returns minimum value of the DataFrame/Row/column.	
max()	Returns max value of the Data Frame/ Rows/ columns.	
sum()	Returns total of value of the Data Frame/Rows/Columns.	

Mathematical Operations on DataFrame:

Pandas offers add(), sub(), mul(), div() functions to manipulate a data frame with scalar values or another data frame. Arithmetic operators (+,-,*,/,//,% etc) can also be used.

Manipulating DataFrame with scalar (Vector Operation): All elements will be manipulated with values.

```
import pandas as pd
dct={'Name':['Amar','Akbar','Anthony','Manpreet'],
     'Phy':[60,65,70,67], 'Chem':[34,55,32,46],
     'Maths':[45,56,65,75]}
df=pd.DataFrame(dct,index=['One','Two','Three','Four'])
print(df)
#Mathematical Operation on a column
df['Chem']=df['Chem']-5
df['Maths']=df['Maths'].add(5)
print(df)
```

	Name	Phy	Chem	Maths
One	Amar	60	39	40
Two	Akbar	65	60	51
Three	Anthony	70	37	60
Four	Manpreet	67	51	70

	Name	Phy	Chem	Maths
One	Amar	60	34	45
Two	Akbar	65	55	56
Three	Anthony	70	32	65
Four	Manpreet	67	46	75

Importing & Exporting Data Frame:

The data of Data Frame can be saved as a file on storage media, so that it can be utilized by other applications. The Data Frame can be saved in different formats but CSV type is simple and commonly used.

The **Comma Separated Value (CSV)** is a simple plain text format which can be created in any Text Editor (Note Pad), Spreadsheet (MS Excel) etc., where all values are separated by comma (,).

Pandas offers the following two methods to import and export data from/to CSV file.

read_csv(): reads/imports data from CSV file to Data Frame.

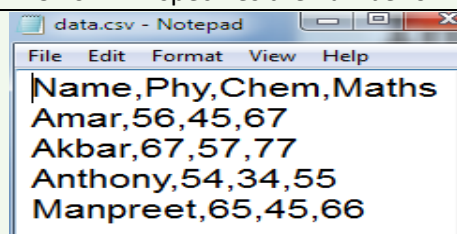
to_csv(): Creates/exports data of Data Frame to CSV file.

Importing (Reading) data from CSV file:

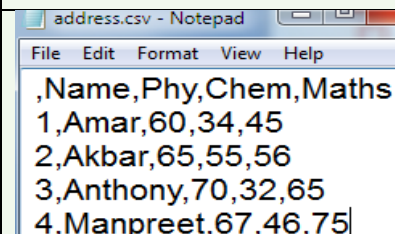
```
<DF>=pandas.read_csv(<"path-filename">, sep=<"Separator">,
                    skiprows=<n>, nrows=<n> )
<"path-filename">: specifies the filename with path. Path name must be
separated by \\. Ex: "c:\\Data\\address.csv"
sep=<"Separator">: specifies the delimiter character. Default is comma.
skiprows=<n>: specifies the number of lines to be skipped while reading.
nrows=<n>: specifies the number of lines to be read from CSV file.
```

Example:

```
import pandas as pd
df=pd.read_csv("e:\\data.csv")
print(df)
```



	Name	Phy	Chem	Maths
0	Amar	56	45	67
1	Akbar	67	57	77
2	Anthony	54	34	55
3	Manpreet	65	45	66



Exporting (Writing) Data to CSV file:

```
<DF>.to_csv(<"path-filename">, sep=<"Sep char"> )
<"path-filename">: specifies the filename with path
(location) to be created. Ex: "c:\\Data\\address.csv"
sep=<"Separator char">: specifies the delimiter character
to separate data values. Default is comma(,).
Row labels & column labels are also written in CSV file.
```

```
import pandas as pd
dct={'Name':['Amar','Akbar','Anthony','Manpreet'],
     'Phy':[60,65,70,67],'Chem':[34,55,32,46],
     'Maths':[45,56,65,75]}
df=pd.DataFrame(dct, index=[1,2,3,4])
print(df)
df.to_csv("e:\\address.csv")
```


Data Visualization:

Data visualization means graphical or pictorial representation of the data using some pictorial tools like graph, chart, etc. Visualization helps to effectively communicate information to users. For example Traffic symbols, Atlas or map book etc. are pictorial representation of facts which are more easy to understand than text.

The **Matplotlib** library of Python, is used for creating various types of 2D- plots in Python. The **pyplot** module of Matplotlib library can be imported in the Python program to plot graphs. **import matplotlib.pyplot as plt** Where plt is name given to pyplot object. You can give any valid identifier/name.

Pyplot methods for plotting graphs:

The following methods are used to plot a chart.

Chart	Method	Description
Line chart	plot()	Visualize data as a series of data point (markers) connected by line.
Bar chart	bar()/barh()	Visualize data with bars with height/ length proportional to value.
Histogram	hist()	Visualize the number of data (frequency) that lie within range of values.

PyPlots also offers some **additional functions** to customize plots.

Method	Purpose
grid()	Configures Gridlines in the graph/plot.
legend()	Displays legend of the axis.
savefig()	Saves the plot as image/pdf file types.
show()	Displays the plot/graph.
title()	Defines the title for the plot/graph.
xlabel()	Sets the label for x-axis.
ylabel()	Sets the label for y-axis.
xticks()	Sets the tick location and label on x-axis
yticks()	Sets the tick location and label on y-axis

Creating Line Chart:

Pyplot's plot() function is used to generate line chart for given values for x-axis and y-axis. Data for X and Y axis may be list, series, column of DataFrame.

```
<pyplot obj>.plot([X-Value,]<Y-Value> [,Line/Marker-color code] [,linewidth=<n>]
[,linestyle=<style code>] [,marker=<marker style>][,markersize=<n>]
[,markeredgcolor=<colorcode>] [,label=<text>])
```

X-Value : Dataset for X-Axis. Default value is [0..N-1], if not given values for X-Axis.

(Dataset may be **1-D Array, List, Series** or **Column of DataFrame**.)

Y-Value : Dataset for Y-Axis. Number of values for X-Axis and Y-Axis should be equal.

Line/Marker-Color: Defines color code for line and Marker symbol.

Linewidth=<n>: A number indicating thickness of line.

Linestyle=<style> : Line style may be **solid, dashed, dashdot** or **dotted** etc.

Marker=<style> : Defines Marker style code.

Markersize=<n> : Defines size of marker in number.

Markeredgcolor=<color code> : Defines color for marker edge.

Label=<text>: Defines label text for legend.

Color Code: r (Red), g (Green), b (blue), k (Black) etc.

Line Style: Solid (-), Dashed line (--), dotted (:) etc.

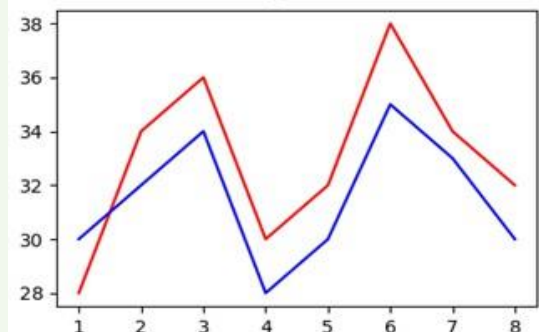
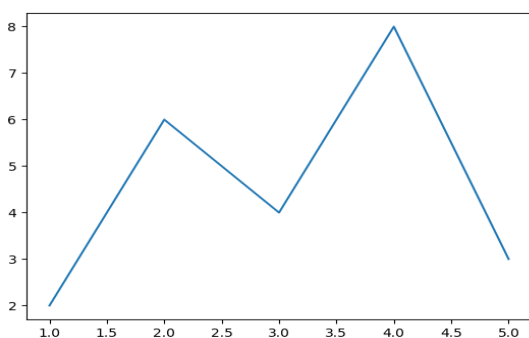
Marker style: Point(.), Circle(o), Plus(+), Cross(x), diamond(d) etc.

creating simplest line chart using List of values

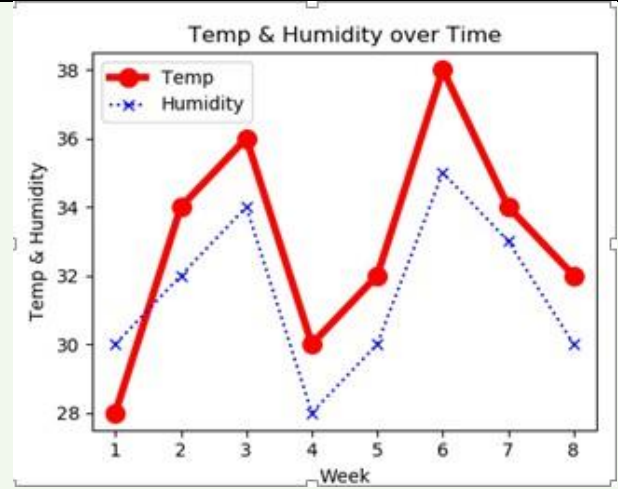
```
import matplotlib.pyplot as plt
xdata=[1,2,3,4,5]
ydata=[2,6,4,8,3]
plt.plot(xdata,ydata)
# Displaying Graph
plt.show()
```

Multi line graph Week and average temp graph

```
import matplotlib.pyplot as plt
w=[1,2,3,4,5,6,7,8]
t=[28,34,36,30,32,38,34,32]
h=[30,32,34,28,30,35,33,30]
plt.plot(w,t,'r')
plt.plot(w,h,'b')
# Displaying Graph
plt.show()
```



```
# line plot with customization
import matplotlib.pyplot as plt
w=[1,2,3,4,5,6,7,8]
t=[28,34,36,30,32,38,34,32]
h=[30,32,34,28,30,35,33,30]
plt.plot(w,t,'r', linewidth=4, marker='o',
         markersize=10, label="Temp")
plt.plot(w,h,'b', linestyle=':', marker='x',
         label="Humidity")
# setting graph
plt.xlabel("Week")
plt.ylabel("Temp & Humidity")
plt.title("Temp & Humidity over Time")
plt.legend(loc="upper left")
plt.show()
```



Creating Bar Chart:

Bar chart is a graphical display of data through colored bars of different height/length. It can be drawn horizontally or vertically. Pyplot's barh() function can be used to plot horizontal bar graph.

<pyplot obj>.bar(<X-Val>,<Y-Val> [,width =<size(s)>][color=<colorcode(s)>] [label=<text>]

X-Value : Dataset for X-Axis. Dataset may be **1-D Array, List, Series** or **Column of DataFrame**.

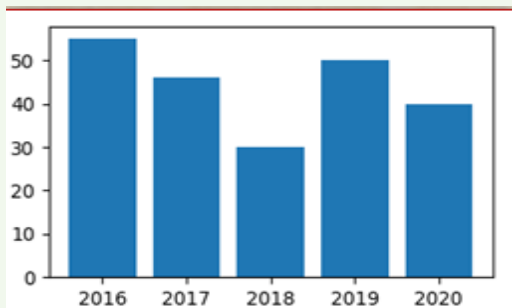
Y-Value : Dataset for Y-Axis. Number of values for X-Axis and Y-Axis should be equal.

width=<size(s)> : Defines width of bars in number. You can also define a list of sizes for each bar, if required.

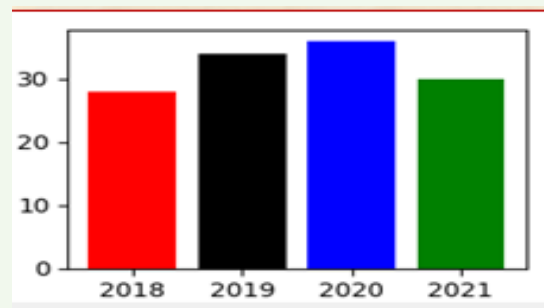
color=<color code(s)>: Defines color for bars. You can also define a list of separate color code for each bar

Label=<text>: Defines label text for legend.

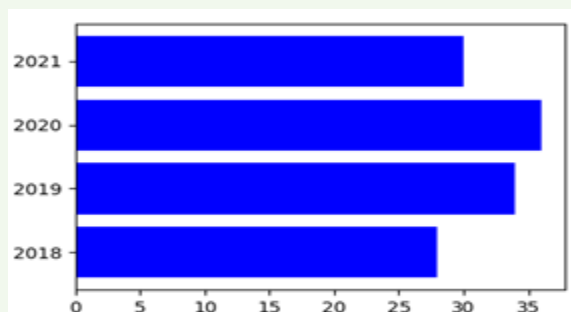
```
# creating simple bar graph using Lists
import matplotlib.pyplot as plt
x=[2016,2017,2018,2019,2020]
y=[55,46,30,50,40]
plt.bar(x,y)
plt.show()
```



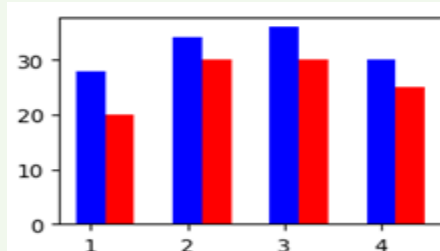
```
# bar chart with varying colors
import matplotlib.pyplot as plt
year=[2018,2019,2020,2021]
pas=[28,34,36,30]
plt.bar(year,pas, color=['r','k','b','g'])
plt.show()
```



```
# Horizontal bar chart
import matplotlib.pyplot as plt
year=[2018,2019,2020,2021]
pas=[28,34,36,30]
plt.barh(year,pas, color='b')
plt.show()
```



```
# bar chart with multiple bar
import matplotlib.pyplot as plt
import numpy as np
x=np.array([1,2,3,4])
male=[28,34,36,30]
female=[20,30,30,25]
plt.bar(x, male, width=0.3, color='b')
plt.bar(x+0.3, female, width=0.3,color='r')
plt.show()
```



Creating Histogram:

A histogram is graphical visualization of the distribution of numerical Data in terms of frequency count. It represents frequencies as bars based on non-overlapping ranges called bins.

Bins can be considered as consecutive, non-overlapping intervals of ranges. It can be a number or range with lower and upper limit.

For example, if bin is 5 then it shows that all data is classified and counted in 5 equal bins or groups.

But if it is given as list of numbers, then it shows ranges. `bins=[0,4,8,12,16]` shows 0-3,4-7,8-11,12-16 ranges. By default, it uses 10 bins with auto-calculated ranges. However, number of bins or ranges can be given as per need. Using `hist()` function we can plot and customize histogram like bins or ranges, histogram types and orientation etc.

`<pyplot obj>.hist(<Data-Values> [, bins =<Number/list of numbers>] [histtype=<type>] [cumulative=<True/False>] [orientation=<'horizontal'|'vertical'>])`

Data-Value : Dataset for the chart. Dataset may be **1-D Array, List, Series** or **Column of DataFrame**.

bins=<number(s)> : Defines the number or bins (in number) and range if given as list of numbers.

histtype=<type>: Defines the type of plot. Type may be bar, barstacked, step and stepfilled. Default is bar type.

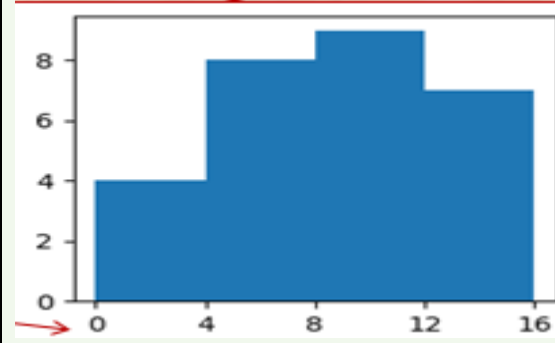
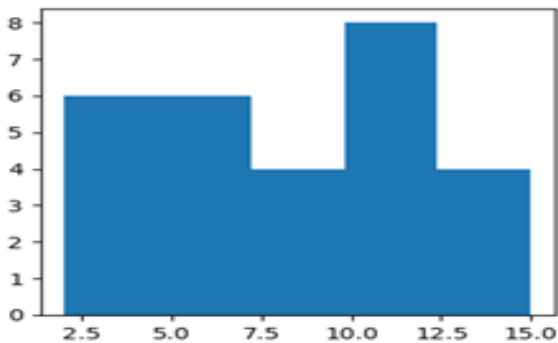
The barstacked is used for multiple data sets in which one is stacked on top of other.

cumulative=<T/F>: Creates cumulative graph if True. Default is false.

orientation=<'horizontal'|'vertical'>: Defines orientation of plot as horizontal bars or vertical bars.

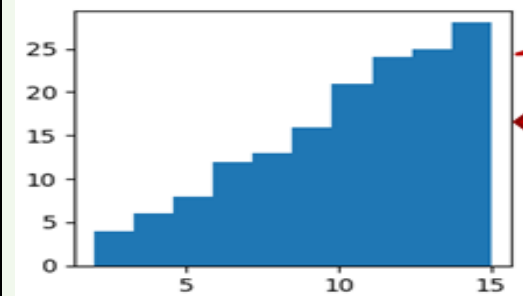
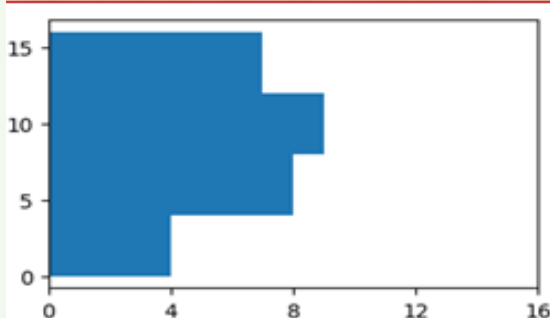
```
# Histogram with 5 bins for age of students
import matplotlib.pyplot as plt
data=[2,4,5,6,2,6,8,10,12,12,11,
10,4,3,2,5,7,9,11,12,13,14,
15,14,10,9,7,9]
plt.hist(data,bins=5)
plt.show()
```

```
# Histogram with 4 ranges and defined x-ticks
import matplotlib.pyplot as plt
data=[2,4,5,6,2,6,8,10,12,12,11,
10,4,3,2,5,7, 9,11,12,13,14,
15,14,10,9,7,9]
plt.hist(data,bins=[0,4,8,12,16])
plt.xticks([0,4,8,12,16])
plt.show()
```



```
# Horizontal Histogram
import matplotlib.pyplot as plt
data=[2,4,5,6,2,6,8,10,12,12,11,
10,4,3,2,5,7, 9,11,12,13,14,
15,14,10,9,7,9]
plt.hist(data,bins=[0,4,8,12,16],
orientation='horizontal')
plt.show()
```

```
# Cumulative Histogram for age of students
import matplotlib.pyplot as plt
data=[2,4,5,6,2,6,8,10,12,12,11,
10,4,3,2,5,7,9,11,12,13,14,
15,14,10,9,7,9]
plt.hist(data, cumulative='True')
plt.show()
```



Saving Plot:

To save any plot use `savefig(<file name>)`

File type may be .png, .jpg, .svg or .pdf.

```
plt.savefig ('myline.pdf')
plt.savefig ('myline.svg')
plt.savefig ('myline.png')
```

UNIT-2 [25 Marks]	1 M	2 M	3 M	4 M	5M	Total
Database Query using SQL	6	4	6	4	5	25 M

What is Database?

A database is an organized collection of interrelated data stored together to serve applications. It work like a container which may contains various database objects.

Most of the databases store data in the form of Relations (also called Tables). Such Databases are known as **Relational Database**: A Software used to store and manage Relational database is called **RDBMS** (Relational Database Management System).

Example of RDBMS software: **Oracle, MySQL, MS SQL Server, SyBase and Ingress** etc.

Advantages of using Database:

- Database reduces **Redundancy**: Removes duplicity of data because data are kept at one place.
- Database controls **Inconsistency**: Data updation maintains consistency and unambiguity.
- Database facilitates **Sharing of Data**: The data stored in the database can be shared among several users.
- Database ensures **Security**: Data are protected against disclosure to unauthorized users.
- Database maintains **Integrity**: DBMS enforces integrity rules to insure the validity or correctness of data.

Relational Data Model:

Data model describes 'How data is organized or stored' in the database. A Database can be categorized in various models as per organization of records like **Relational Data Model, Network Data Model, Hierarchical Data Model** and **Object Oriented Data Model**.

Relational Data Model: In Relational model, data is organized in the form of Relation or Table consisting rows and columns. This model is mostly used by RDBMS software.

Relation: A Relation or Table is Two-Dimensional structure containing in Rows and Columns. It has the following properties-

- **Column homogeneous**- All values in a column are of same data type.
- **Unique columns**- Each column assigned a unique name and must have atomic (indivisible) value.
- **Unique rows**- All rows of a relation are distinct i.e. no two identical records can exist in the Relation (Table).

Tuple/Entity/ Record: A Row of a table is called Tuple or Record.

Attribute/ Field: A column of a table is called Attribute or Field.

Domain: It is collection (set) of possible values from which the value for a column is derived.

Degree: Number of columns (attributes) in a table.

Cardinality: Number of Records in a table.

Concept of Keys:

A table may have several columns and some **column** or **combination of columns** which can identifies a record in the table uniquely. Such column(s) are called **Key** of the Table. Keys can be categorized in following types.

Primary Key: A column or group of Column, which uniquely identify a Record/Tuple in a Table is called Primary Key. The Primary key cannot have null or duplicate values. **A table can have only ONE primary key**. Example: Vehicle Number, Enrolment number, Admission No, Roll No, PNR Number , Aadhaar , PAN No. etc.

Candidate Key: A Table may have multiple keys having candidature to work as primary key. Collection of all such key-columns is called candidate keys. Note that Primary key is one of the candidate key.

Alternate Key: All candidate key other than Primary key are called Alternate key (Secondary Key) because they can be used as an alternative to Primary key.

Foreign Key: is a **non-key attribute**, which establish a relation with another table. Mostly it is common column in two tables and works as primary key of another table.

Consider the following tables and their columns:

EMPLOYEE (EmpNo, Name, Designation, City, AadharNo, DeptNo)

DEPARTMENT (DeptNo, Dname, HODName)

Candidate Keys of EMPLOYEE Table : (EmpNo, AadharNo)

Primary Key of EMPLOYEE Table- EmpNo & Primary Key of DEPARTMENT Table- DeptNo

Foreign Key of EMPLOYEE Table- DeptNo. (because it is common to both table and is primary key in Department Table)

MySQL:

MySQL is an open source Relational Database Management System (RDBMS) software based on SQL. The main features of MySQL are –

- **Open Source & Free:** It is Open Source and available free of cost.
- **Portability:** It can be installed and run on any types of Hardware and OS like Linux, MS Windows or Mac etc.
- **Security:** It offers security and authorization feature to keep database secure.
- **Connectivity:** The MySQL database can be connected with Programming Languages like Python, Java etc.
- **Query Language:** It uses SQL (Structured Query Language), which is standardized by ANSI.

Types of SQL (MySQL) Commands:

SQL commands may categorised into two category as their usage:

DDL (Data Definition Language): Commands to define structure of data i.e. Database and Table.

DML (Data Manipulation Language): Commands to process records stored in the table.

DDL Commands (Works with Table/Database)	
CREATE	Creates Database and Tables
ALTER	Modifies structure of Table
DROP	Deletes Database and Table

DML Commands (Works with Records)	
SELECT	Displays records from the table.
INSERT	Insert records in the table
DELETE	Deletes records from the table
UPDATE	Modifies values of existing records.

Apart from these commands, MySQL the following miscellaneous commands –

SHOW DATABASES: Displays list of Databases available.

SHOW TABLES: Displays list of tables stored in currently opened database.

USE <Database Name>: Opens a database for working. Example : USE EMPLOYEE

DESCRIBE <Table>: Display structure of table. Example – DESCRIBE Student

Data Types in MySQL:

A column of the Table stores same type of values as defined while creating a table. Commonly used MySQL data types for a column are-

INTEGER (Size)/INT (size) : Stores numbers without decimal upto given size.

DECIMAL (M, D)/NUMERIC (M, D)/FLOAT (M,D) : Numbers with decimal points upto M size and D decimal places.

Ex: Decimal (8,2) will store like 123456.78

CHAR (size): Stores string/text upto given size. Maximum size is 255 and default is 1, if size not given.

VARCHAR (size): Stores variable size string upto given size. Maximum size is 65535.

DATE: Stores date string in YYYY-MM-DD format.

TIME: Store time string in HH:MM:SS format.

Note: All values must be enclosed in single or double quotes except Integer and Decimal numbers.

Working with DDL Commands

Creating Database and Table:

CREATE DATABASE <Data base Name>

CREATE TABLE < Table Name> (<Col name1><data type>[(size)] [Constraints],....);

Constraints:

Constraints are the rules, condition or checks applicable to a column or table which ensures the integrity or validity of data. The commonly used constraints are-

Constraints	Purpose
NOT NULL	Ensures that a column cannot have NULL value.
UNIQUE	Ensures that all values in a column are different.
DEFAULT	Provides a default value for a column, when nothing is given.
CHECK	Ensures that all values in a column should satisfy certain condition.
PRIMARY KEY	Defines Primary Key of the Table that can identify a row uniquely.
FOREIGN KEY	Defines Foreign key which makes a relation with any other table to ensure Referential Integrity of the data.

Note: Constraints are optional. A table may have multiple Unique constraints but only one Primary Key.

Creating Database and table.

CREATE DATABASE SCHOOL;

USE SCHOOL;

CREATE TABLE Employee

(Code char(3),

EmpName char(20),

City varchar(40),

Pay Numeric(8,2));

Creating table with constraints.

CREATE TABLE Student

(StCode char(3) NOT NULL PRIMARY KEY,

Stname char(20) NOT NULL,

StAdd varchar(40),

AdmNo char(5) UNIQUE,

StSex char(1) DEFAULT 'M',

StAge integer CHECK (StAge>=5));

<p>Modifying Structure of Table: We can alter (modify) the structure of existing table by the using ALTER TABLE Command. With ALTER TABLE Command, we can-</p> <ul style="list-style-type: none"> • Add a new Column or Constraints • Modifying existing column (data type, size etc.) • Delete an existing column or Constraints • Rename Column Name <p>ALTER TABLE <Table Name> ADD MODIFY DROP CHANGE <Column Definition(s)></p>	<p>Adding new column - ALTER TABLE Student ADD (TelNo Integer); ALTER TABLE Student ADD (Code int(3) Primary Key);</p> <p>Modifying Existing Column : ALTER TABLE Student MODIFY Name VARCHAR(40);</p> <p>Deleting Column: ALTER TABLE Student DROP TelNo;</p> <p>Renaming Column Name: ALTER TABLE EMP CHANGE ENAME EMPNAME CHAR(40); OR ALTER TABLE EMP RENAME ENAME EMPNAME;</p>
<p>Deleting Database and Table: DROP DATABASE <Data base Name> DROP TABLE <Table Name></p>	<p>DROP DATABASE School; DROP TABLE Student;</p>
<p>Working with DML commands</p>	
<p>Inserting Records: You can insert record in the table by using by using the following DML command. INSERT INTO <Table Name> [<Column list>] VALUES <list of values></p> <p>Note:</p> <ul style="list-style-type: none"> ▪ Values to be given as per order of columns. ▪ 'Null' can be used in place of unknown values. ▪ You can define order of column with table name as per order of values to be inserted. 	<p>Suppose a Table STUDENT (Code, Name, Fname, DOB, Class) is created. The following command will insert a record.</p> <p>INSERT INTO Student VALUES ('s1','Amar', 'Dhirendra Kumar','1985-10-25', 12); INSERT INTO Student VALUES ('s2','Shaan', NULL,'1972-5-25', 10); INSERT INTO Student (StID, FName, Name, Class) VALUES ('s3','Amitabh', 'Abhishek', 12);</p>
<p>Deleting Records: DELETE FROM <Table Name> [WHERE <Condition>]</p>	<p>DELETE FROM Student ; (All record will be deleted) DELETE FROM Student WHERE City='Mumbai' ; DELETE FROM Student WHERE Class >=11 ;</p>
<p>Modifying Records: UPDATE <Table> SET <Column> = <Expression> [WHERE <Condition>]</p>	<p>UPDATE Student SET Class =10 ; UPDATE Student SET Class=10 WHERE class=9 ; UPDATE Emp SET Sal = Sal+(Sal*10/100) WHERE Sal <=10000; UPDATE Emp SET City = 'Dehradun' WHERE CITY IS NULL;</p>
<p>Displaying Records: The SELECT command used to make a request (query) to retrieve stored records from the table. SELECT < [Distinct ALL] * column name(s)> FROM <table(s)> [WHERE <condition>] [ORDER BY <column(s)> [ASC DESC]] [GROUP BY <column(s)> [HAVING <condition>]] ;</p> <p>A condition defined with WHERE clause may have the following Relational and Logical operators: =, >, <, >=, <=, <>, IS, LIKE, IN, BETWEEN, OR, AND, NOT (!)</p> <p>Note:</p> <ul style="list-style-type: none"> ▪ IS operators is used to compare NULL value (Never use = (equal) with Null. ▪ BETWEEN operator defines a range of values as lower and upper range (both are inclusive) ▪ LIKE Operator used to define a condition based on a pattern of string. <p>Pattern is a search string having the following wild cards- Percent (%) - Represents a substring in any length Under score (_) - (Represents a single character at used position.)</p> <p>Example.</p> <ul style="list-style-type: none"> • 'A%' : Any string starting with 'A' character. • ' __A' : Any 3 character string ending with 'A'. • '_B%' : Any string having second character 'B' • ' ___ ' represents any 3 letter string. 	
<p>Making Simple Queries Using SELECT Clause:</p>	
<p>Displaying all records with columns SELECT * FROM Student ;</p> <p>Displaying all records with selected columns. SELECT Name, City FROM Student ;</p> <p>Displaying unique values in a column SELECT DISTINCT City FROM Student ;</p>	<p>Using Column Aliases - SELECT Name, DOB AS 'Date of Birth' FROM Student;</p> <p>Making Expressions: SELECT 4*3 ; SELECT Name, Sal*12 FROM EMP; SELECT Name, Sal*12 AS 'Annual Salary' FROM EMP;</p>

Displaying Selected Records- WHERE SELECT * FROM Student WHERE City='Mumbai'; SELECT Name, City from Student WHERE City <> 'Mumbai' AND Class>10; SELECT * FROM Emp WHERE Sal >10000 OR Job = 'Manager'; SELECT * FROM Student WHERE NOT Grade='A';	Searching NULL Values – IS Operator SELECT * FROM Student WHERE City IS NULL ; SELECT * FROM Student WHERE City IS NOT NULL;
---	---

Specifying Range of Values – BETWEEN Operator SELECT * FROM Emp WHERE Sal BETWEEN 5000 AND 10000; [Same as SELECT * FROM Emp WHERE Sal >= 5000 AND Sal <= 10000 ;] SELECT * FROM Emp WHERE NOT Sal BETWEEN 5000 AND 10000 ;	Specifying List of values- IN Operator SELECT * FROM Emp WHERE Sal IN (5000, 10000) ; [Same as SELECT * FROM Emp WHERE Sal = 5000 OR Sal = 10000 ;] SELECT * FROM Student WHERE City IN ('Mumbai', 'Delhi', 'Kanpur') ;
---	--

Specifying Pattern of string- SELECT * FROM Student WHERE Name LIKE 'A%'; SELECT * FROM Student WHERE Name LIKE '%Singh%'; SELECT Name, City FROM Student WHERE Class >= 9 AND Name LIKE '%Kumar%';	Ordering result of query – Order By Clause: SELECT * FROM Student Order By City; SELECT * FROM Student Order By City DESC; SELECT * FROM Student WHERE City="Prayagraj" Order By Name;
---	---

Functions in MySQL:
A function is a special types of command in MySQL that performs some operation on table and returns a single value as a result. MySQL functions may be divided in the following categories-

- **Math or Numeric Functions** (works on numeric columns and numeric data)
- **Text or String Functions** (works on string data or text columns)
- **Date & Time Function** (works on date and time data or columns)
- **Aggregate Functions** (works on numeric/Text columns vertically and also called Multi Row function)

Numeric Functions:		SELECT MOD(5,2) ; → 1 SELECT POW(5,2) ; → 25 SELECT ROUND('212.567',1) ; → 212.6 Select ROUND(15.193,1); → 15.2 Select ROUND(15.193); → 15 Select ROUND (355.35,-1) → 360 Select ROUND (124.35,-1) → 120 Select ROUND (355.35,-2) → 400 SELECT TRUNCATE('212.567',1) ; → 212.5 SELECT ROUND(PAY,2) FROM EMP; SELECT TRUNCATE(PRICE*QTY,2) FROM ITEMS;
MOD(N,M)	Returns remainder of N divide by M	
POWER(N,M) POW(N,M)	Returns N to the power M (N ^M)	
ROUND (N [,M])	Returns a number rounded off up to M place. If M is -1, it rounds nearest 10. If M is not given, the N is rounded to the nearest Integer.	
TRUNCATE(N,M)	Returns number after truncating M decimal places.	
SQRT (N)	Returns square root of N	

String Functions:		SELECT LENGTH('abcd') ; → 4 SELECT LENGTH(Name) FROM Student; SELECT LOWER('ABCD') ; → abcd SELECT LOWER(Name) FROM Student; SELECT UPPER('abcd') ; → ABCD SELECT UPPER(Name) FROM Student; SELECT LTRIM(' abcd') ; → abcd SELECT LTRIM(Name) FROM Student; SELECT RTRIM('abcd '); → abcd SELECT TRIM(' abcd '); → abcd SELECT SUBSTR('MY COMPUTER', 4,3') → COM SELECT LEFT('MYSQL', 2) → MY SELECT LEFT(Name, 4) FROM Student; SELECT RIGHT('MYSQL', 3) → SQL SELECT MID('COMPUTER', 4,3) → PUT SELECT MID (Name, 4,3) FROM Student; SELECT SUBSTR('ABCDEFG', -5, 4) ; → CDEF SELECT SUBSTR('ABCDEFG', 3); → CDEF SELECT INSTR('CORPORATE', 'OR'); → 2
LENGTH(str)	Returns the length of given string	
LOWER(str)/ LCASE(str)	Returns given string in lower case.	
UPPER(str)/ UCASE(str)	Returns given String in upper case	
LTRIM(str) RTRIM(str) TRIM(str)	Removes Leading/Trailing/both spaces from given string.	
LEFT(str, N) RIGHT(str,N)	Returns the (N) characters from left/right from the given string.	
SUBSTR(str,P,[N]) MID (str,P,N)	Returns the substring for given position (P) and (N) characters. If N is (-ve) then backward position counted.	
INSTR(str1,str2)	Returns the index of first occurrence of str2 in str1.	

Date & Time Functions:	
CURDATE() CURRENT_DATE()	Returns the current date in YYYY-MM-DD format.
NOW()	Returns the current date & Time as YYYY-MM-DD HH:MM:SS
DATE()	Returns the date part of a date-time expression.
DAY(), MONTH(), YEAR()	Returns the Day/Month/Year from given date or date type column.
DAYNAME()	Returns the name of the weekday
DAYOFMONTH()	Returns the day of month (1-31).
DAYOFWEEK()	Returns the day of week (1-7).
DAYOFYEAR()	Returns the day of year (1-366).

```

SELECT CURDATE() ; → 2024-10-31
SELECT CURDATE()+10; → 2024-11-10
SELECT NOW() ;
SELECT DATE('2008-12-31 01:02:03') ;
      → 2008-12-32
SELECT YEAR('2008-12-31') ; → 2008
SELECT YAER(DOB) FROM Student;
SELECT MONTH('2008-12-31') ; → 12
SELECT * From Emp Where MONTH( DOB)=10;
SELECT * FROM Student Where DAY( DOB)=15;
SELECT DAYOFWEEK('2008-12-31') ; → 1
SELECT DAYOFYAER('2010-02-05') ; → 36

```

Aggregate Functions:	
SUM()	Returns the sum of given column.
MIN()	Returns the minimum value in the column.
MAX()	Returns the maximum value in the column.
AVG()	Returns the Average value of the given column.
COUNT()	Returns the total number of NOT NULL values in the column/ number of records.

Note:

- Aggregate functions works on whole column (vertically) and ignores NULL values.
- Never use Aggregate functions with WHERE clause.

```

SELECT SUM (Sal) FROM Emp;
SELECT SUM(Sal) FROM Emo WHERE
City='Guwahati';
SELECT MAX (Sal) FROM Emp;
SELECT MAX(Sal) FROM Emp WHERE
City='Jaipur';
SELECT AVG (Sal) FROM Emp;
SELECT COUNT (Name) FROM Emp;
SELECT COUNT (*) FROM Emp;
SELECT COUNT(*) FROM Emp
WHERE City='Jaipur';
Note: Count(*) gives number of records.

```

Grouping Records - GROUP BY Clause	
<p>Some time it is required to apply a Select query <u>on a group of records instead of whole table.</u></p> <p>GROUP BY <column> [HAVING <Condition>] clause with Select command makes groups. A group column is chosen which have non-distinct (repeating) values like City, Job etc.</p> <p>Condition may be applied on Groups with HAVING. Generally, the Aggregate Functions are applied on groups.</p> <p>Tips: A query question containing “wise” or “for each” indicates the use of Group By clause with Select.</p>	<pre> Select Sum (Sal) from EMP where City='Kanpur'; Select Min (Sal) from EMP Group By City; Select Job, Sum(Sal) from EMP Group By Job; Select AVG(Sal) from EMP Group By City; Select Count(*) from EMP Group By City; Select Job, Sum(Pay) from EMP Group By Job HAVING Avg(Pay)>=7000; Select Job, Sum(Pay) from EMP Group By Job HAVING Count(*)>=5; </pre>

Joining two tables (Join Query):
<p>Sometimes it is required to access information from two or more tables, which requires the Joining of tables. A query in which two or tables are required to join is called Join Query. The following major types of Join can be implemented on tables –</p> <p>Cross Join: Each record of Table-11 is joined with each record of Table-2. It is also called Cartesian product. If Table-1 has 3 columns and 5 records, and Table-2 has 4 columns and 6 records then 5x6=30 records with 3+4=7 columns will be generated as result.</p> <p>Natural Join: Tables are joined on the equality of common column and common column will be displayed once.</p> <p>Equi-Join: It is mostly similar to Natural Join where two tables are joined on the equal value of given column but record contains all column from both tables.</p> <p>The commonly used method of Joining table is-</p> <p>SELECT < Column(s)> FROM <Table1, Table 2 > WHERE <Joining Condition> [Order By ..] [Group By ..]</p> <p>You may add more conditions using AND/OR/NOT operators, if required.</p> <p>Join condition mostly usage Common Column of both tables and written as-</p> <p>Table1.Common col=Table2.Common col (You can also use alias (short name) of tables in Join condition.)</p> <p>Ex. Find out the name of Employees working in Production Deptt.</p> <p>Select Ename From EMP, DEPT Where Emp.DeptNo=Dept.DeptNo AND Dname='Production';</p> <p>Ex. Find out the name of Employees working in same city from where they belongs (hometown).</p> <p>Select Ename From EMP, DEPT Where Emp.DeptNo=Dept.DeptNo And City=Location;</p> <p>Ex. Find out the name of students enrolled in B.Tech Course.</p> <p>Select Ename From STUDENT AS S, COURSE AS C Where S.CourseID=C.CourseID and Cname='B.Tech';</p>

UNIT-3 [10 Marks]	1 M	2 M	3 M	4 M	5M	Total
Introduction to Computer Networks	3	2	-	-	5	10 M

A computer network is a collection of two or more computing devices which are interconnected to share data and other resources. The Computer network facilitates transfer or exchange of information in the form of text, image, and audio, video through wired or wireless transmission medium.

A computer Network may include-

- **Hosts or Nodes:** Devices receives or sends data like computers, laptops, mobiles etc.
- **Network devices:** Devices manages transfer of data like Switch, Router and Modem etc.
- **Transmission Media** (wired or wireless): Media which carries data signals like wire or Bluetooth etc.
- **Protocol:** Set of rules which control the communication.

Advantages of Computer Network:

- **Sharing Resources:** Network facilitates sharing of hardware and software resources like sharing of data, programs and printer etc. among users irrespective of their physical location.
- **Improved Communication:** Computer network enables fast, reliable and secure communication between users like Sending e-mail, SMS and MMS etc.
- **Reduced Communication cost:** Sharing resources also reduces its communication cost. Internet and Mobile network playing very important role in sending and receiving text, image, audio and video data at low cost.

Types of Computer Network:

A network may vary in size, complexity and coverage area. On the basis of coverage of geographical area, data transfer speed and complexity, a computer network may be classified as:

- **LAN (Local Area Network) :** LAN connects devices placed in limited geographical area like a single room, a floor, buildings or campus. LAN usage wires (Ethernet cables) or wireless (Wi-Fi) to connect devices and offers high speed data transfer rates usually varies from 10-100 Mbps.
- **MAN (Metropolitan Area Network):** The MAN may connects several LANs and covers a larger geographical area like a city or a town. Cable TV network or cable based broadband internet services are examples of MAN. This kind of network may be extended up to 100 km.
- **WAN (Wide Area Network):** The WAN covers large geographical area like countries and continents. A WAN is formed by connecting several LANs and MANs. The Internet is the largest WAN that connects billions of computers, smart phones and millions of LANs from different continents.
- **PAN (Personal Area Network):** The PANs are small network which connects devices in small proximity up to 10 meters using wired USB connectivity or wireless system like Bluetooth, Infrared , Wi-Fi etc. PAN facilitates transfer of files like songs, videos, images etc. from one computer, mobile to other.

	PAN	LAN	MAN	WAN
Covered Area	Approx. 10 mt.	Room, Building, complex or campus	May cover a city	Country or continents.
Media used	Data cable, Infrared/Bluetooth.	Ethernet , Coaxial cable, Wifi etc.	Optical fiber, Radio wave, Microwave	Microwave and satellite etc.
Network Devices used	Cable, Dongle	LAN card, Hub, Switch, Repeater	MODEM, Router, Gateway	MODEM, Router, Gateway

Transmission Media:

A Transmission medium is a medium of data transfer over a network. The selection of media depends on the **cost, data transfer speed, bandwidth** and distance.

Transmission Medium is divided into two major categories:

(A) Wired (Guided) Media:

- **Twisted Pair Cable:** Twisted pair or Ethernet cable is most common media which consists of four insulated pairs of wires twisted around each other. It is low-cost, flexible and easy to install cables and can transfer data upto 100 mts distance. It uses RJ-45 Connector for connecting devices.
- **Co-Axial Cable:** This cable consists a solid insulated wire surrounded by wire mesh separated by foil or insulator. Co-axial Cable or **Coax**, is mostly used in Cable TV transmission and can carry data upto 500 mts.
- **Optical Fiber:** An optical fiber is a thin, flexible, and transparent strand of glass or plastic that carry light signals instead of electric current. Signal are transmitted in the form of light emitted from source using Light Emitting Diode (LED) or LASER beam. Optical fibers offers secure and high speed transmission upto a long distance.

(B) Wireless (Unguided) Media:

- **Infrared Wave:** It used for short-range (approx. 5 mt) communication using wireless signals. It is mostly used in Remote operated devices like TV, Toys, Cordless phones etc.
- **Bluetooth:** Bluetooth is a wireless technology for creating personal networks operating within a range of 10 meters.
- **Wi-Fi (Wireless Fidelity):** Wi-Fi communication is similar to Bluetooth in operation, but it covers a large coverage (50-200 mts.)
- **Radio waves:** Radio wave is used to make broadcast network like FM Radio within city. Radio wave propagates in Omni direction (surrounding) and can penetrate solid walls/buildings etc.
- **Microwaves:** Microwave are high energy electromagnetic waves propagates in line of sight direction. It is high speed wave and can cover distance upto 100 km.
- **Satellite:** Satellite works like a Trans-Receiver Antenna in the space, which receives, regenerates and redirects signals using Microwave. Services like DTH, VSAT, GPS and Satellite phones etc. are offered by the satellite.

Network Devices:

Networking devices are equipment that receive or transmit data or signal and make communication channel. Some common Networking devices are-

- **Network Interface Card (NIC):** A NIC (Network Interface Card) or LAN Card enables computer to connect with a network using a RJ-45 port. Each LAN card possess a unique **6 Byte** Physical (static) address known as **Media Access Control (MAC) Address**, which is used to identifies a device uniquely over the network. **WLAN** (Wireless) card is also used for connecting PC/Laptops with Wireless Network.
- **Hub:** A Hub is a device which connects multiple computers together to form a Local Area Network (LAN). Hub makes Broadcast type Network and do not manage traffic over the network channel. Nowadays Hub is obsolete technology and Switch is used in place of Hubs.
- **Switch:** Switch is similar to Hub that also connects several nodes to form a Network. But Switch is faster than hub due to better traffic management and control over Network. Switch can also be used to combine various small network segments to form a big Network.
- **Repeater:** A Repeater is a device that regenerates the received signals and re-transmits to its destination. In case of twisted pair cable (Ethernet), signals become weak after 100 meters. So, Repeaters are required at each 90- 100 meters to maintain signal strength. A Hub or Switch also works as a repeater.
- **Router:** Router is an inter-networking device which connects multiple Networks to form a big Network. The basic role of Routers is to determine the best possible route (shortest path) for the data packets to be transmitted. In a large network (WAN), multiple routers work to facilitate speedy delivery of data packets.
- **Gateway:** A Gateway device connects dissimilar networks having different protocols. Gateway is also called protocol converter, since it converts data packets from one protocol to other while connecting two dissimilar networks. Usually it is implemented by software within a router device.
- **MODEM (MODulator-DEMulator)** device connects Telephone line to the Computer. It converts Digital signal into Analog (Modulation) and Analog to Digital (Demodulation). This conversion is required because Telephone lines can't carry digital data. Generally it is used to connect a PC with Telephone lines to access Internet or make voice call using PC.

Network Topologies:

The layout of interconnection of computers and devices in a network is called Topology.

The selection of topology for a network, depends on Cost of media, Flexibility to add more devices and Reliability of network (Fault detection in case of Network failure). Some commonly used Topologies are-

Bus Topology:

In the bus topology, all devices are connected to a main or backbone cable. It is simple and oldest topology used in the early days of computer networking.

Ring Topology:

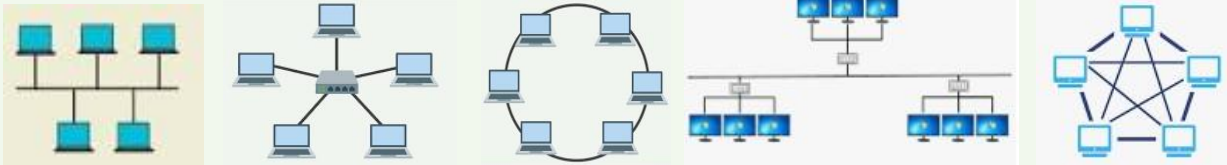
In a ring topology, all nodes are connected to a main cable making a closed loop (ring). All data packet travel in the ring in the same direction and passes through each node.

Tree Topology:

Tree topology combines multiple star topology networks together onto a bus (Bus-Star approach). In its simplest form all connecting devices (hub or switch) are connected to the bus network to make "root" of the network tree.

Mesh Topology:

In Mesh topology, all devices are directly connected to each other, forming a mesh-like structure. Mesh topology is most expensive and complex to implement and manage.



Introduction to Internet:

- **Internet** is a **network of networks** that consists of millions of private, public, academic, business, and government networks, that are linked by various wired, wireless, and optical networking technologies.
- The **Internet** is a **global system of interconnected computer networks** that use the standard Internet protocol suite (TCP/IP) to serve billion users worldwide.
- The modern Internet is an extension of **ARPANet** (Advanced Research Project Agency Network), created in 1969 by the American Department of Defense.
- The Internet carries variety of information resources and services, such as the inter-linked hypertext documents of the World Wide Web (WWW), communicational infrastructure to support e-mail, chat and transfer of Text, Images, Audio, Video etc.

Application of Internet:

□ World Wide Web (WWW):

World Wide Web (WWW) or Web is a collection of inter-linked hypertext pages accessed through Web Browser program using internet. A web page may contains information in form of text, images, audio, video or Animation. The resources on the web can be shared or accessed through the Internet. The Web pages (HTML pages) are stored on web server and transferred via the Hypertext Transfer Protocol (HTTP) using a software application called a web browser.

Tim Berners Lee, a British computer scientist invented the revolutionary World Wide Web in 1990 by defining three fundamental technologies i.e. HTML, WWW and URL.

- **HTML:** Hyper Text Markup Language (HTML) is a language which is used to design Web Pages. A web page is designed using HTML tags like <HEAD>,<BODY> etc. to define the web page layout and organized contents which to be displayed by the web browser. Webpages are stored as .html or .htm files.
- **URL (Uniform Resource Locator) :** URL is a unique address of web resources located on the web. It provides the location and mechanism (protocol) to access the web resource. URL is also called a web address. A URL contains protocol, subdomain, Domain and name of webpage along with path/directory.



Protocol- Protocols may be http, https, ftp and telnet etc.

Subdomain- Subdomain may be www, blog, and mail etc.

Domain- Domain contains name of website and 3 characters Top level domain (TLD) like .com, .edu, .org, .net etc. Sometimes country domain (2 letters) also used like .uk, .in, .nz etc.

Path and Web page- last segment of URL may contains location/path and webpage name.

□ E-Mail:

Email (electronic mail) is the ways of sending and receiving message(s) using the Internet. The message can be either text onto the email application or an attached file (text, image, audio, video, etc.). E-mail address like abc@xyz.com identifies the sender and recipients. Now days Web-based e-mail services are available at free of cost through various service providers like Google-Gmail, Yahoo- Yahoo mail, Microsoft-Hotmail & Outlook etc. E-mail service providers may provide extra facilities like filtering spam, search emails by sender or contents and organizing contacts and email ids, sending email to cc, bcc (blind carbon copy) recipients etc.

An Email application uses SMTP (Simple Mail Transfer Protocol), IMAP (Internet Mail Access Protocol) and POP(Post Office Protocol) etc. to handle mails over network.

❑ **Instant Messaging (Chat):**

Instant Messaging (IM) or Chatting over the Internet means communicating to people at different geographic locations in real time through text message(s). It is similar to e-mail, except that message is sent immediately to recipient. It facilitates user to type and send messages to make conversation. Applications such as **WhatsApp, Slack, Skype, Yahoo Messenger, Facebook Messenger** etc., are examples of instant messengers. Some applications support sending audio and video along with text chat.

❑ **Voice Over Internet Protocol (VoIP):**

Voice over Internet Protocol (VoIP), allows make voice call over the Internet. VoIP offers voice transmission over a computer network (IP) rather than through regular telephone network. It is also known as Internet Telephony. Examples of Voip are Whatsapp, Skype, Google Chat etc.

❑ **Video calls and Video Conferencing :**

Video Conferencing allows two or more people to make two-way video and audio transmissions. In Video conferencing, group of people at multiple locations may participate rather than individuals. **H.239** is commonly used protocol for Video conferencing. Google Meet, Microsoft Team, Skype, Zoom or WhatsApp are most commonly used applications for video conferencing or meeting.

Website:

In general, a website is an organized collection of interlinked web pages related through hyperlinks, stored on a web server to provide information or service to its visitor.

All the pages of a website are stored under one domain name and have a common theme or template. For example, the website of CBSE will have all the pages related to syllabus, Circulars, Events, resource materials etc., under one domain name and having a common design theme.

A website can be accessed by providing the address of the website (URL) in the browser. The main page of website (**Home/Index page**) will be open when it is opened on the browser.

Web Page:

A web page is a HTML document on the WWW that can be viewed through web browser. To make web pages more attractive, various styling CSS (Cascaded Style Sheet) and formatting are applied on a web page. Further, program codes (scripts) may also be used to make webpage interactive and define different actions and behavior. **JavaScript, PHP** and Python are commonly used script language.

A web page is usually a part of a website and may contain information in the forms of texts, images, audio and video etc. Depending on the functionality and features, a web page may be classified as-

Static Web Page:

A static webpage is one who's content always remains static, i.e., does not change for person, location or device. Static web pages are generally written in HTML or CSS and have the extension .htm or .html.

Dynamic Web Page:

A dynamic web page is one in which the content of the web page can be different for different users according to language, location or device. The dynamic are more complex in design and take more time to load than static web pages. Dynamic web pages can be created using script languages such as JavaScript, PHP, ASP.NET, Python etc. Web pages displaying the date, time, weather report, multi-lingual pages or auto resizing webpage as per screen size of the client device are Dynamic web page.

Static V/s Dynamic Web Page

Static Webpage	Dynamic Web page
The static web pages display the same content each time, irrespective of location or device.	The dynamic Web page changes its appearance or contents according to user, location or device.
It takes less time to display on browser at client end.	Dynamic web pages take more time while displaying on browser.
These are simple in layout and style and no Database is used.	Mostly database is used at server-end to store contents and style information.
Mostly designed through HTML and stored as .htm or .html format	Designed in Script language like Java Script, ASP.Net, PHP etc. and stored as .php,.asp etc.

Web Server:

A web server stores and delivers the contents of a website to web clients (browser) when requested.

A web server as a computer, stores Web server Application software and a website's contents (HTML pages, images, CSS style sheets, and Script files).

Web server as a software, is a specialized program that understands URLs and requests from browsers, and

responds to those requests.

Each web server is assigned a unique domain name, so that it can be accessed from anywhere using Internet. The web browser from the client computer sends a HTTP request for a page and web server then accepts request, interprets, searches and responds (HTTP response) against request of the web browser. The requested web page is then displayed in the browser of the client. If the requested web page is not found, web server generates "Error: 404 Not found" as a response.

Web Hosting:

Web hosting is a service that enables you to publish website on the Internet. Once a website is created, it should be placed on web server for global access.

For webhosting, we need storage/space on a web server where all the files (webpage) and necessary data can be stored. Mostly Web Server services (CPU, RAM, and storage etc.) are taken on rent basis from the cloud service providers and web developer hosts website on server by uploading the files constituting the website (HTML, CSS, JavaScript, images, databases, etc.). These services are usually paid and resources can be increased or decreased as per requirement.

All web servers are assigned a unique numeric address called IP address when connected to the Internet. This IP address needs to be mapped to domain name (Website name/URL) of the website using DNS (Domain Name Service). The domain name can be registered (purchased) through an authorized agency i.e. Registrar Domain Names.

Web Browser:

A Web browser is a software application that helps us to view the web page(s) from web server to any client device. Some commonly used web browsers are **Google Chrome, Internet Explorer, Mozilla Firefox, Opera**, etc. A web browser essentially displays the HTML documents which may include text, images, audio, video and hyperlinks that help to navigate from one web page to another.

The modern browsers allow a wide range of visual effects, use encryption for advanced security and also have cookies that can store the browser settings and data.

Every web browser offers certain settings that define the manner in which the browser will behave. These settings may be related to privacy, search engine preferences, download options, auto fill and autocomplete feature, theme and much more. Each browser application allows customization of its settings through setting options.

Add-ons and Plug-ins:

Add-ons and plug-ins are the tools that help to extend the functionality of the web browser.

An add-on is also called extension in some browsers and is used to add a particular functionality to the browser like playing sound or graphics is an example of an add-on. Mostly these Add-ons are provided as Extensions through a library provided by the Web Browser which can be installed from the browsers setting options.

A **plug-in** is a complete program or third-party software which offers some extra functionality to users. For example, Flash players and Java are plug-ins etc. A Flash player is required to play a video in the browser. A plug-in is a software that is installed on the host computer and can be used by the browser for multiple functionalities and can even be used by other applications as well.

Cookies:

A cookie is a small text file, which is transferred by the webserver to the browser while surfing a website. The information stored in cookies are related to user, pages were visited, choices and menu(s) settings on a particular website. It helps in customizing the web pages like language, auto-login information, preferences and sometimes remembering the shopping preference and displaying advertisements of interest, etc.

Cookies are usually harmless and they can't access information from the hard disk or transmit virus or malware. However, cookies may be privacy threats. Sometimes viruses can also be tricked as cookies and cause harm to a computer system. Cookies can be disable by changing the Privacy and Security settings of the browser.

UNIT-4 [10 Marks]	1 M	2 M	3 M	4 M	5M	Total
Societal Impacts	3	4	3	-	-	10 M

Digital Footprint:

Digital footprint is the trail of information that people leave online or in any communicating devices.

Digital footprint includes all activities carried over digital devices and Internet like websites we visit, emails and messages we send, and any information we submit online, posts on social media, online purchases, software and apps used, music, video downloaded or uploaded etc.

All our online activities leave a data trace on the Internet as well as on the computing device that we use. This can be used to trace the user, their location, device and other usage details.

There are two kinds of digital footprints we leave behind.

- **Active Footprints**

Active digital footprints includes information that we intentionally submit online. This may include emails we write, or responses or posts we make on different websites or mobile Apps, etc.

- **Passive Footprints**

The digital data trail we leave online unintentionally is called passive digital footprints. This includes the data generated when we visit a website, use a mobile App, browse Internet, etc.

Net and Communication etiquettes:

Now days our society is turning as Digital Society. Digital society reflects the growing trend of using digital technologies in all spheres of human activities. Anyone who uses digital technology along with Internet is a **digital citizen** or a **Netizen**.

Being a good netizen, all of us need to be aware of some ethics, morals and values to be maintained during online. A responsible netizen must abide by the following etiquettes-

Net etiquettes :

Net Etiquettes or Netiquettes is code of good behavior on the Internet. It may include-

Be Ethical	<ul style="list-style-type: none"> ▪ No copyright violation: We should not use copyrighted materials without the permission of the creator or owner. While downloading and using text, image, audio, video from the internet, we must mention courtesy or credit to original author or owner. ▪ Share expertise: Share only true and unambiguous information and knowledge on Internet so that others can access to get benefitted from it.
Be Respectful	<ul style="list-style-type: none"> ▪ Respect privacy: We should respect the right to privacy and the freedom of expression of others. Our personal communication may include images, documents, files, etc., that may be private to both. We should not share such images, documents, files, etc. without consent of owner. ▪ Respect diversity: In a group or public forum, we should respect the diversity of the people in terms of knowledge, experience, culture, Religion and other aspects. Avoid comments and passing remarks on others in social media platforms.
Be Responsible	<ul style="list-style-type: none"> ▪ Don't feed troll: Writing comments with the aim of getting a reactions from others is commonly referred to as "trolling". Since trolls thrive on attention, the best way to discourage trolls is not to pay any attention on such comments. ▪ Avoid cyber bullying: Any insulting or degrading behavior like posting of rumors, giving threats online, posting the other's personal information, sexual harassment or comments aimed to publicly ridicule a person is called cyber bullying. Cyber Bullying is a punishable offence and our actions can be traced using our digital footprints.
Be Secure	<ul style="list-style-type: none"> ▪ Secure password: We should have strong and frequently changed password. Never share personal credentials with others to avoid misuse of account. ▪ Know who you befriend: Be careful while being friend to unknown people as their intentions possibly could be malicious and unsafe. ▪ Beware of fake information: We should apply our knowledge and experience to validate Fake news, messages and posts. Never forward fake posts.

Communication Etiquettes:

Digital communication includes email, texting, instant messaging, talking on the cell phone, audio or video conferencing, posting on forums, social networking sites, etc. We should follow the following etiquettes while making communication-

Be Precise:

Always respect other's precious time and data usage. So while writing emails, messages etc. be concise in contents and share large sized files as attachment to avoid data usage for download. However necessary large files may be shared through cloud storage like Google Drive, OneDrive, Dropbox etc.

Be Polite:

We should be polite and non-aggressive in our communication. So avoid being abusive even if we don't agree with others' point of view.

Be Credible:

We should be cautious while making a comment, replying or writing an email or post. Such acts decide our credibility and reflect our digital personality.

Data Protection:

Data protection refers to the practices, safeguards, and rules to protect your personal information and ensure that you remain in control of it. Personal data refers information relating to your private life like biometric information, health and financial information, or other personal documents, images or audios or videos etc.

How to protect our private Data?

- Be careful while sharing information on Internet.
- Browse websites privately and ensure that the URL contain HTTPS and a pad lock sign.
- Ensure safe and trusted sites while entering sensitive information.
- Never save passwords while working on public computer.
- Avoid entering sensitive information onto a public computer and erase history and traces of your work from public computers, if used.

Intellectual Property Rights (IPR):

Intellectual Property refers to the literary and artistic works, inventions, designs and symbols, names and logos etc. The ownership of such property lies with the creator, or the holder of the intellectual property.

The owners of Intellectual Property holds certain exclusive legal rights through which he/she can use their property without any disturbance and can prevent the misuse of their property. Like other tangible property, owner of IPR can earn recognition or financial benefit by using their creation or invention. So, violation of intellectual property right is punishable offence.

In India, Intellectual Property is governed under the Patents Act, 1970; Trademarks Act, 1999; Copyright Act, 1957; Designs Act, 2001 etc.

Intellectual Property is also legally protected through **copyrights, patents, trademarks** etc. commonly known as Intellectual Property Right (IPR).

Copyrights:

- Copyright gives legal rights to creators for their original works like **writing, photograph, audio recordings, video, sculptures, architectural** works, **Software** and other creative works like **literary** and **artistic** work.
- The copyrights include right to copy (reproduce), share, display, distribute etc. It prevents others from copying, using or selling the work.
- Copyrights are automatically granted to creators and authors. Copyright law gives the copyright holder a set of rights that they alone can avail legally. Copyright is valid for life time of the author and 60 years after his/her death.

Patent:

- A patent is usually granted for inventions, innovations and process etc.
- Patent owner gets an exclusive right to prevent others from using, selling, or distributing the protected invention and gives full control to the patentee to decide whether or how the invention can be used by others.
- The inventor needs to apply (file) for patenting the invention. A patent protects an invention for 20 years.

Trademark:

- Trademark includes any visual symbol, word, name, design, slogan, label, etc., that distinguishes the brand or commercial enterprise, from others.
- Trademark prevents others from using a confusingly similar mark, including words or phrases. For example, no company other than Nestle can use the Nescafe brand name to sell coffee.
- The owner of a trademark may take legal action against trademark infringement. Most countries require formal registration of a trademark for product or services for 10 years, if not renewed.

Plagiarism:

Plagiarism is stealing someone else's intellectual work, idea or thoughts and representing it as one's own. Plagiarism is an unethical practice of using other's work without giving credit to its original author or source. Sometimes considered as an act of fraud. Some common examples of plagiarism are-

- Copying someone's work and presenting it as one's own.
- Act of stealing or copying literary work and not giving the credit to its author or original creator.
- Copying software programs written by other programmers and claiming them as one's own.

The following points may be used to avoid plagiarism.

- Always use your own idea and words. Never claim other's work and idea as your own.
- Always give reference of sources or website, if material is taken from elsewhere.
- Always give acknowledgement and credit to author or creator, if his/her material is used.
- Quotations or phrases should be used as it is with citation of author.

Difference between Plagiarism and Copyright:

Plagiarism	Copyright
Plagiarism is using someone else's work or ideas without giving proper credit.	Copyright infringement includes the unauthorized or unlicensed copying of a work.
Plagiarism is a violation of academic norms and unethical practice but normally not illegal.	Copyright violation is a crime and punishable illegal offence in terms of law.
Plagiarism applies when ideas are copied	Copyright violation occurs only when a specific content like sequence of words or image is copied.

Licensing and copyright:

A license is a type of contract or agreement between user and creator or copyright owner to describe the terms under which user is allowed to use the copyrighted material.

Generally licensing term is related to computer software when developer/author of the software share their works with others under **license**, and allows others to **use** and **even modify** the content. Since, software are also intellectual work and are subject to copyright, so software are used under certain usage guidelines and restrictions imposed by developer or software company. These usage guidelines are called License.

The Licensing is different from copyright in the following ways-

Licensing	Copyright
License is an agreement between user and content developer which describes the usage rights and restrictions on its application.	Copyright is legal right which describes the use and access of contents without transferring ownership of the contents.
It describes the terms under which people are allowed to use the copyrighted material.	Copyright is the legal term used to declare and prove who owns the intellectual property.

Free and Open Source Software (FOSS):

In general, open source refers to any program whose source code is made available for use or modification by others i.e. Source code is open for all. Open source software is usually developed as a public collaboration and made freely available to all without any restrictions.

Open Source Software (OSS) which are freely available to all for self-use, study, distribute to others and modification or improvement in the software are called Free and Open Source Software (FOSS).

Example: OpenOffice, Linux, Java, PHP, MySQL, Python etc. are commonly used Open Source Software.

Cybercrime and Cyber laws:

Cybercrime refers to any crime wherein the computer is either a tool or a target or both.

Some examples of Cyber Crime are-

- Posting offensive messages on Social Networking Portals.
- Hacking of Computer or Cracking Security systems.
- Unethical Financial transactions and Fraud through Internet
- Harassment through e-mails and messages.
- Creation & propagation of Virus, Worms or Trojans etc.
- Cyber terrorism.

Like traditional crime such as theft, fraud, forgery, defamation and mischief, Cybercrime are also treated as criminal activities and are subject of punishment. The **Information Technology Act 2000 (IT Act)** in India provides protection against cybercrime and also known as Cyber Law.

Hacking:

Hacking is the act of unauthorized access to a computer, computer network or any digital system.

A hacker gains unauthorized access to computers or networks in order to steal sensitive data with the intent to damage or bring down systems. Hackers as technical expert of the hardware and software, and use their skill for illegal and malicious purposes or personal gain.

An Ethical hackers explore any vulnerability or loophole in the system with positive intent and reports to the owner for improvement.

Phishing:

Phishing is an act of acquiring sensitive and private information such as User name, passwords, Credit card number, bank account details etc. using a trap-mail or voice call in which user himself discloses their private details. The most common phishing method is through a fake or forged email in which user is asked to provide confidential information and the user presumes that it is being asked by authentic source and gives all details in good faith.

Phishing attempts through phone calls or text messages are also common these days. So never disclose your private and confidential data to anyone without knowing its authenticity.

Cyber Bullying:

Cyberbullying is a form of harassment using electronic means or social media. Harmful bullying behavior can include posting rumors, threats, vulgar remarks, revealing victims' personal information etc.

Victims of cyberbullying may experience lower self-esteem and various negative emotional responses, including being scared, frustrated, angry, or depressed. Cyber Bullying is different from Troll in which a person (troll) posts inflammatory, insincere, digressive, or off-topic messages in an online community with the intent of provoking readers into displaying emotional responses, or manipulating others' perception.

Overview of Indian IT Act:

The **Information Technology Act 2000 (IT Act)** in India provides legal framework against cybercrime and e-commerce activities. The Information Technology Act, 2000 (IT Act) was notified on **17 October 2000**. The original Act contained **94 sections, divided into 13 chapters** and 4 schedules.

The IT Act is applicable to all parts of the country. Persons of other countries can also be indicted under this Act, if the crime involves a computer or network located in India.

The IT Act has been amended to cope with new challenges on 23rd December 2008 (IT Act 2008) and notified on October 27, 2009. IT Act–2008, is a new version of IT Act 2000.

E-waste: hazards and management:

- E-waste (Electronic waste) includes electric or electronic gadgets and devices that are no longer in use. E-waste includes discarded or unusable electrical-electronic devices like computers, laptop, tablets, mobile phones, television sets, and other electronic home appliances.
- E-waste is one of the fastest growing environmental hazards in the world. It is major cause of degradation of environment and pollution.
- When e-waste is carelessly thrown or dumped in landfills or grounds, certain chemicals and metals used in electronic products cause air, water and soil pollution. Harmful chemicals and poisonous gases mixed up in groundwater, soil, and air and therefore affect animals and human life through crop's yields, drinking water and breathing.
- If e-waste is not disposed of in proper manner, it can be extremely harmful to humans, plants, animals and the environment. Therefore, it is very important that e-waste is disposed of in such a manner that it causes minimum damage to the environment and society.

Managing e-Waste:

We can minimize the impact of E-waste by efficient disposal of e-waste. The best policy for e-waste management is **reduce, reuse** and **recycle**.

Reduce: Try to reduce the generation of e-waste by maximum use of devices till its useful life through proper maintenance.

Reuse: The working equipment may be donated or sold to someone who is still willing to use it.

Recycle: Recycling is the process of conversion of e-waste into something that can be used again and again in some or the other manner.

Disposal of e-Waste:

- Donate your old usable and working devices to needy people, who can use it.
- Old and obsolete devices may be returned manufacture or seller to purchase new one (Buy Back).
- Non-repairable devices may be given to Recycling Agencies for proper disposal or re-cycling.
- Non-functional devices may be donated to Service Centers, so that they can utilize some functional parts while repairing others.
- Do not through electronic product in open area or sell to street garbage collectors because they have no proper disposal system.

Awareness about health concerns related to the usage of technology:

Today's life cannot be imagined without using technologies like Mobile, Internet and other IT tools. The modern technologies have effected every domain of human life like communication, education and health. Technology has transformed our lives, and for most of us, life may not be easy without using gadgets and technologies.

But Excessive usage of digital devices has a negative impact on our physical as well as psychological well-being.

Some common impacts are-

- Decreasing use of our own memory.
- Decreased use mental and physical activity.
- More Device dependency.
- Addicted to screens, social media, games etc.
- Poor physical health: Eye strain, Muscle Problems, Sleep disorder, gaining weight etc.
- Social and emotional issues: Isolation, Depression, anti-social behaviour etc.

Proper use of technologies and devices:

- Use devices in proper posture. Prefer an ergonomic furniture and follow natural sitting style.
- Limit your time and set boundaries of using technologies. Never be addicted to Internet and Social media.
- Follow 20-20-20 rule i.e. take a 20-second break from the screen every 20 minutes and look at something 20 feet away to minimize eye strain.
- Take regular breaks from technology to reduce stress and improve mental clarity.
- Take help of others when needed and avoid excess use of technology and dependency on gadgets.